SYSTEMS ENGINEERING RESEARCH CENTER

Agile Systems Engineering – Kanban Scheduling Subsection

Technical Report SERC-2017-TR-104

March 10, 2017

Principal Investigator: Dr. Richard Turner, Stevens Institute of Technology

Co-Principal Investigator: Dr. Alice Smith, Auburn University

Research Team:

Auburn University:
Dr. Jeffrey Smith
Donghuang Li
Gokhan Ozden

Stevens Institute of Technology:
Paul McGary

University of Southern California:
Alexey Tregubov

Sponsor: DASD(SE)

The Systems Engineering Research Center (SERC) is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contract HQ0034-13-D-0004-0059.

Any views, opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense nor ASD(R&E).

No Warranty.
This Stevens Institute of Technology and Systems Engineering Research Center Material is furnished on an "as-is" basis. Stevens Institute of Technology makes no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material. Stevens Institute of Technology does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

TABLE OF CONTENTS

LIST OF FIGURES

EXECUTIVE SUMMARY

Goals of this research:

Develop the Demonstration and Analysis Tool for Agile SE Management (DATASEM)—a flexible modeling and simulation capability to:

1. Enable **realistic experiments** to understand how **governance models**, organizational structures and work flows interact across a system of systems
2. Provide a framework to **calibrate assumptions of performance**
3. Integrated **experiment generation tools**

DATASEM is intended as an initial instantiation of an evolving and expanding set of integrated tools to support research and transition.

Results of RT-159 activities:

1. The suite of software developed and delivered in December of 2015 was determined to have fundamental defects that caused it to improperly represent the concepts as originally intended
2. The defects were largely caused by incomplete or ambiguous definitions of several of the model mechanism concepts
3. A more definitive description of the concepts was created and delivered in a technical report
4. A data model describing information produced by the simulation was developed to support the new descriptions
5. There were insufficient resources to complete the development of the suite to align with the refined definitions
6. An experiment based on data from an aerospace industry source was defined as a MS project by a Stevens graduate student; the stand-alone version of the software was modified sufficiently to incorporate mechanisms to support the student's experiment
7. Updated software is available through www.sercuarc.org
8. One journal article and one conference paper were published. A second conference paper will be produced based on the results of the experiment.

Next Steps:

- The software suite will continue to be evolved using resources outside the SERC
  - The MS student will produce a paper on the results of his experiment.
  - A simulation result analysis tool (simulation playback) is being developed at USC
  - Graduate student researchers will continue to refine the system to support their dissertation research where appropriate
- Other research areas are identified

The modified software and all ancillary documentation will be available to anyone on the SERC website (www.sercuarc.org). Questions may be addressed to any of the authors.

INTRODUCTION

BACKGROUND

Developing, creating or evolving systems of systems (SOSs) present significant systems engineering and management problems. Dahmann and Baldwin characterize these problems as stakeholder involvement, governance, operational focus, acquisition, test and evaluation, boundaries and interfaces, and performance and behavior [1]. All systems face some of these problems, but the uniqueness of the dynamics and resulting communication issues in a SoS require a significant ability for adaptation within the system development community, as well as among the stakeholders. The principles for addressing these issues are no different from those required for any good systems engineering and development activity [2]. Implementation of those principles in SoS environments, however, is a much thornier problem.

Agile and lean philosophies have shown to be effective in supporting adaptation within development and evolution [3], [4], [5]. Complicated, large systems of systems in rapid or continuous deployment environments, where requirements are not precise and can change or emerge quickly, find traditional approaches inadequate.

In 2011, the Systems Engineering Research Center began to investigate alternative management and governance approaches for these complex environments, including a concept for an integrated multi-level network of pull scheduling systems based on explicit, transparent, and continuously updated value of work [6], [7], [8]. This Kanban-based Scheduling System Network (KSSN) concept was developed based on the following capabilities:

- Coordinate multiple levels of development activity across multiple system components with diverse and possibly disjoint or isolated development groups
- Support analysis and decision making at every level
- Flexibly schedule work considering value across the system of systems
- Balance work in progress (WIP) across resources with SoS organizational capacity to improve flow
- Make visible to all levels progress toward capability development and deployment
- Establish a basis for continuous improvement in a rapidly changing environment

Difficulties in validating this concept *in vivo* led to the decision to create a broad simulation environment that would allow in vitro experimentation with KSSN, but also be applicable to studying other mechanisms, singly and in concert, operating in a range of organizational structures (including all four types of systems of systems identified in [1]) and handling different kinds, durations, complexity, and volumes of work flow. We believe that establishing statistically significant evidence across various combinations of mechanisms, organizations and work flows, as well as providing a suitable simulation "sandbox" for adopters to perform their own experiments will provide a level of confidence that *in vivo* experimentation (piloting) is low risk and provides value to adopters.

OBJECTIVES

DEVELOP THE DEMONSTRATION AND ANALYSIS TOOL FOR AGILE SE MANAGEMENT (DATASEM)

The Demonstration and Analysis Tool for Agile SE Management (DATASEM) is a flexible modeling and simulation capability to advance the understanding of the KSSN value-based concepts, to investigate optional mechanisms for implementation, and provide support for organizations that are interested in piloting the concept. DATASEM will support broader, in-vitro experimentation required to provide comparative information across a broad set of implementation architectures and organizations as well as store information from *in vivo* pilots. Additionally, it will graphically demonstrate the key concepts of these adaptive management approaches to interested organizations.

RELATIONSHIP TO PREVIOUS SERC RESEARCH

This research builds upon previous findings from three earlier SERC research tasks:

- MPT, *Evaluation of Systems Engineering Methods, Processes and Tools on Department of Defense and Intelligence Community Programs*, derived an initial methodology for evaluating software-related MPTs that might be applicable in systems engineering through surveys and literature searches.
- RT-35/35A, *Agile-Lean Software Engineering (ALSE) Evaluating Kanban in SE*, focused on using pull scheduling techniques to determine the applicability of Kanban scheduling to systems and software engineering in a rapid response environment. It also introduced the possibility of systems engineering as a service.
- *RT-124, Agile Enablers and Quantification*, identified and evaluated potential mechanisms that might be worthwhile to simulate with DATASEM.
- *RT-126, Agile Systems Engineering – Kanban Scheduling* developed an initial suite of tools and documentation including both online and standalone versions.

RESEARCH GOALS

The overall Agile SE Management Project research goals are to:

1. Identify agile, lean, and other adaptive processes and governance mechanisms to help systems engineers
   a. Identify, analyze and quickly react to issues in an environment of accelerating change
   b. Keep pace with evolving requirements, risks and opportunities throughout the extended development lifecycle
   c. Understand and manage the changing economic and political factors that undergird and enable system development
   d. Broaden SE influence and holistically approach complications from increasing
      i. Creation and evolution of systems of systems
      ii. Interoperability between legacy and new capabilities
      iii. Reductionism resulting in point solutions or locally optimized decisions

2. Provide a modeling environment to validate and experiment with adaptive mechanisms, their interactions with more traditional SE, and how they can balance adaptability with discipline in a broad variety of environments.
3. Inform organizations contemplating changes to their system development processes or working in system of systems environments where there are different development approaches being applied concurrently.

Specific goals for this task were:

Organizational Modeling Tools.

> The researchers shall improve the utility of DATASEM tool for building experiments and analyzing results, with the goal of informing decisions regarding the structure of the engineering work to be performed and the organizations doing the work. The researchers shall focus on improving the ease of use for both building experiments and analyzing outcomes. The researchers shall also investigate other organizational modeling tools to identify capabilities that complement DATASEM functionality, such as Stanford's POW-ER tool: Process, Organization, Work for Edge Research.

Calibration and Validation.

> The researchers shall validate DATASEM through a set of rigorous experiments using the experimental validation framework developed under RT-126. The researchers shall compare DATASEM results to data collected from pilot efforts to calibrate and improve the DATASEM capability. The researchers shall investigate whether DATASEM results can be used to gauge the effectiveness of different organizational or work structures, and identify when work estimates are overly optimistic or conservative.

DATASEM is intended as an initial instantiation of an evolving and expanding set of integrated tools to support research and transition.

SUMMARY OF WORK PERFORMED

The RT-159 effort occurred during the 12-month period between 11 March 2016 and 10 March 2017.

INITIAL VALIDATION

While the concept for the mechanisms to be implemented in the DATASEM suite primarily grew out of the results of the RT-35/35a work, it became clear during the final report generation for RT-126 that there were fundamental errors in the DSL and the software functionality. During the first months of RT-159, those errors were identified and analyzed. The result was the realization that there was simply not a unified understanding across the development team of the definitions associated with the governance mechanisms. This was a significant blow to the research, but it was decided that without a common, coherent definition of the mechanisms, the team had little chance to correct the existing defects.

A first attempt at validation of the DATASEM software was also conducted using a different simulation tool (SIMIO) as a reference. Appendix A provides a short white paper on the methodology and the results.

LOOKING FOR ALTERNATIVE MODELS

One idea that came out of the analysis and the concomitant concern regarding our limited resources given that the definitions would use up a good deal of our development resources, was to investigate existing simulation tools. If a suitable tool could be found that met at least some of the needs of the DATASEM, it might be possible to simply add the new mechanisms and not need to redevelop the underlying infrastructure. The primary tool of interest was a commercial version of a simulation developed at Stanford University in collaboration with the Naval Postgraduate School: The Process, Organization, Work for Edge Research (POW-ER) tool. Unfortunately, the commercial tool was specifically oriented toward the risks and uncertainties of schedules. It provided a rich set of risk categories. These would be of some use, but the concepts of flow and value were not specifically addressed. Another barrier identified was the unknown configuration and status of the software that was available for DATASEM extension. The developer indicated that this academic version was written in several languages and would need to be reconstructed. Based on these barriers, the team decided to continue with the existing DATASEM infrastructure.

CREATING THE MODEL AND MECHANISM CATALOG

Significant effort was dedicated to establishing the revised definitions. The work primarily addressed the interactions among the organizational, governance, and work models in terms of the definitions of organizational activities and the governance mechanisms used to implement the activities. The final results were delivered in Technical Report (A013): *Model Catalog and Mechanism Definition*, SERC-2017-TR-159-001, February 17. 2017. This document provides clear definitions of the model components used in DATASEM, the governance mechanisms we are working to understand and investigate, and the general approach and its relevance to industry and research.

A survey-based review of these definitions was designed and successfully navigated through the Stevens and OSD Human Subject Research Internal Review Boards. Unfortunately, when it was finally ready to be released, there was insufficient calendar time to distribute it, collect responses, and analyze the data before the task ended. The instrument is available in Survey Monkey, and will be useful for follow-on research.

A WELL-TIMED EXPERIMENT

In early January, Paul McGary, a Master of Science degree candidate in the School of Systems and Enterprises, contacted the team with a proposal for his final project. He proposal included the following goals.

> *Establish a better technical project management and systems engineering process using self-governing teams involving customer interaction, an agile approach to milestone and schedule development, Kanban resource and task assignment, and the Incremental Commitment Spiral Model (ICSM) to manage constantly changing risks, clarifying objectives and verifying the design approach at every opportunity.*
>
> *To verify this new approach, legacy projects will be analyzed and "re-run" against the DATASEM simulation model presented at the December 2016 SoSCIE Webinar using applied concepts such as organizational components, governance models, and a combination of predictive and adaptive scheduling and work flows to further refine the process described above.*

Two significant issues arose in responding to Mr. McGary's needs. The first was the need to represent both adaptive and pre-defined scheduling mechanisms in the simulation. The second was the impact of the organizational component's technical approach on the definition of the work item network. These issues are closely related in that adaptive approaches tend to create smaller batch sizes and have shorter planning horizons, while pre-defined schedules tend to have larger batch sizes and a longer planning horizon.

The initial (December 2015) DATASEM software was primarily focused on the adaptive approaches. Conversely, it assumed that the work item network was relatively static; that is, the assignment of work items was pre-defined in the DSL. The decision needed to be made as to whether the work item network could be predefined but still allow for simulating organizations with different technical processes and planning horizons.

After much deliberation, the decision was made to identify and make changes to the December 2015 tool necessary to support the experiment.

With less than three months to go before the task ended, this opportunity focused the team on an immediately useful subset of capability while also addressing both goals described in the Research Goals section above. Thus, Mr. McGary's paper will form a significant part of the output of this task. It will be available in May of 2017.

IMPROVING THE USER INTERFACE

One of the goals was to provide a more useful and informative user interface. One task looked at improving the native presentation provided in the stand-alone version through RePast and to support Mr. McGary's experiment. A second task created an initial data model to describe the type of information to capture during a simulation and a format that would be useful in analysis. The data model is being used in the "Simulation analysis tool prototype" project as defined in the Ongoing Research section, and the JSON format defining it is provided in Appendix B.

Outcomes of the Research

The RT-159 products are based on the specific goals of the research and include contractual deliverables, technical reports, software, publications, and conference presentations.

Contract deliverables

Per the statement of work, we delivered:
A005, Funds & Labor Expenditure Report
A008, Bi-monthly status reports
A009, Technical and Management Work Plan
A010, Contractor Roster
Technical Report *Model Catalog and Mechanism Definition*, SERC-2017-TR-159-001, February 17. 2017.
A013, Final technical report summarizing the research findings (this document)

Software

Updated software is provided and available through www.sercuarc.org. The software falls into 6 component categories:

- DSL Editor
- DSL Compiler
- DSL Instantiation
- Simulation
- User Interface
- Results display

The software is available in source code and executable formats. The package contains all the custom and open source software in executable formats with instructions on installation. The web-enabled version of the DATASEM has not been updated.

Non-deliverable Publications and Presentations

- McGary, P., "Agile Systems Engineering and Technical Project Management," Master of Science Project, Stevens Institute, to be completed in May, 2017.
- Smith, Jeffrey, "System of Systems Task Management Decision Support Using Agent Based Modeling and Simulation," presentation to the Institute of Industrial and Systems Engineering Conference, May 22-24, 2016. Anaheim, CA.
- Turner, Richard; Alice E. Smith, Jeffrey Smith, Levent Yilmaz, Donghuang Li, Saicharan Chada, and Alexey Tregubov, "DATASEM: A Simulation Suite for SOSE Management Research," Proceedings from the Eleventh IEEE International Conference on Systems of Systems Engineering (SoSE), Kongsburg, Norway, June 12-16, DOI: 10.1109/SYSOSE.2016.7542954 react-text: 57.
- Turner, R., "The Impact of Agile and Lean on Process Improvement,“ Crosstalk, March, 2017.

Conclusions and Next Steps

This phase of DATASEM evolution has provided insights into the simulation implementation approach and the governance mechanisms, and provided support to an experiment based on industry data. It has also spawned additional research efforts outside of SERC funding.

Insights

The learning cycle is most effective when defects are identified and analysis is performed to correct the errors. The DATASEM development has presented the team with a number of insights directly related to defects in a number of assumptions.

Complexity across Organizational, governance and work models

The most important insight of this and the previous development effort was the level of complexity represented by interactions among the various models. When initially considering the simulation suite, we believed that the three activities we were interested in modeling were essentially independent and that we could easily mix and match different types. As we began to develop the DSL and implement the various mechanisms, it became clear this was not the case.

A key example is the dependence of the actual work item characteristics on the technical process. As described in the Model and Mechanisms Catalog, the Work Item Network generation algorithm that decomposes work items within an operational component (OC) is provided in the Governance Model for the organizational component. We discovered that it needs to vary according to whether the governance for the OC is characterized as adaptive or predictive.

Adaptive technical processes often use multiple iterations or increments with constant or changing cadences to provide continuous value and validation. They may order the WIs according to some value formulation, or to handle resource constraints due to the changing environment of the project. They tend to use smaller work batch sizes to enable more efficient flow. They may enforce broad Classes of Service that are used by other OCs, internal Work In Process limits, or other methods to support better flow. They may also allocate work to other OCs to manage their internal flow.

Predictive technical approaches usually generate (or obtain) a detailed schedule and then execute it as closely as possible. Schedules are redeveloped when the actual work reaches some defined incongruence to the schedule. This redevelopment provides insight and allows management adjustments. However, because of the longer planning horizon and slower adjustment cadence, work items are generally larger, and the decomposition and possible allocation to external OCs becomes less flexible.

The result of this dichotomy is that there needs to be additional complexity in the work item generation algorithm to incorporate a means for describing this predetermined work allocation.

Difficulty in specifying behavior of governance mechanisms

The unexpectedly complex interactions among the models influenced the lack of a common understanding of the specific activities that make up mechanisms we were studying. Even with this realization, making the mechanism definitions more specific led to differing ideas about the actual behavior associated with the mechanisms, and how to represent them in the simulation.

One result of this was a significant change in the way the organizational model is defined. By refining the definitions of the organizational model and using a single Organizational Component (OC) as a building block, we simplified the organizational concept, but increased the number of characteristics that needed to be captured for each OC. An OC could be a complete organization, a contracted entity, a team, or an individual resource. This allowed us to use an OC to represent a systems engineer with specific skills (such as security) and provided that engineer the complete range of OC activities (work acceptance, ordering and resources allocation, internal work execution, and work status monitoring) as well as a governance process to implement those activities. While this is an excellent way to better understand such concepts as systems engineering as a service, it creates additional complexity in the actual description of the OC activities within the governance model.

The interaction also meant that the work description became more complex. OCs need to know additional details about aggregating work items (e.g. capabilities, requirements, etc.) to be able to decompose them. For example, the skills distribution among the subtasks of an aggregating work item requires the user to be more specific in defining the type and expected level of resources.

To help the user deal with all of this additional complexity, we established the concept of experiment model templates. Templates are based on executable DSL models that include pre-specified, consistent definitions, and present the user with a reduced set of user-defined variables. The variables are primarily associated with the number and type of OCs, the size and make up of the work, and the stochastic values used to drive the simulation. An elementary version of this is currently implemented in the DSL definitions through the use of libraries, but the template would not require the user to craft the DSL. Development of a few draft templates was part of the re-definition work, but no functional capability using those drafts was implemented.

Ongoing Research

DATASEM and the Agile SE program have led to additional research outside the SERC. This section describes work ongoing and planned that grew out of the overall Agile Systems Engineering program.

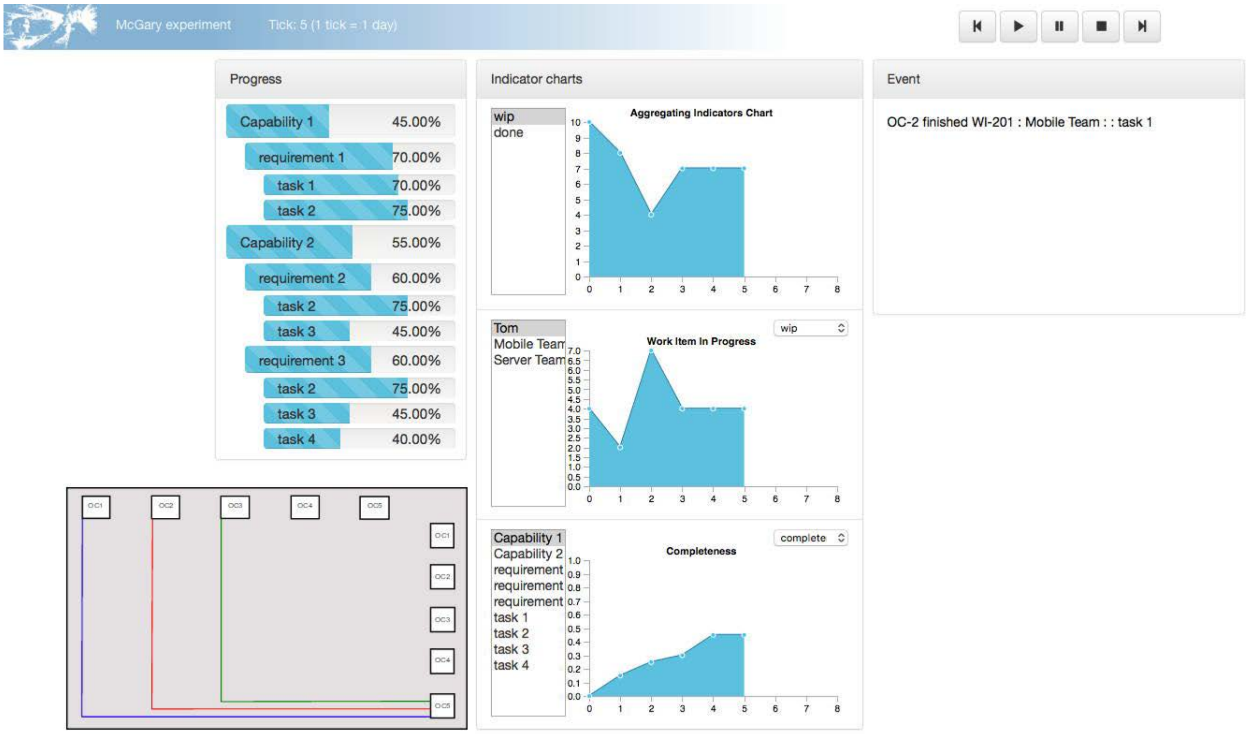
Simulation analysis tool prototype

The full DATASEM suite has always envisioned tools to better visualize and analyze simulation results. The simulation analysis tool prototype, research supported by the Center for Systems and Software Engineering at USC, is developing a simulation playback capability. Simulation results can be reviewed in discrete steps with a continuously tailorable set of visualizations (such as graphs and interaction displays) and an event narrative. Navigation uses transport controls like those used in audio and video players. Figure 1 is a screen shot from an initial prototype of the player. The full prototype is expected to be available in May of 2017.

SIMULATION OF WORK INTERRUPTIONS AND MULTITASKING

It has been observed that multitasking can cause inefficient (or unproductive) work. Modern lean and agile practices in software engineering processes also acknowledge the problem and attempt to eliminate waste by limiting work in progress and using better team organization and work scheduling techniques. Existing research has studied multitasking and work interruptions on individuals, but very few of them evaluate effects of multitasking on the team or the whole organization. The goal of the research is to understand how multitasking and work interruptions affect the cost of projects.

To measure the lasting effect of interruptions DeMarco and Lister introduced a concept of the “reimmersion time” [9], illustrated in Figure 3.



Capability 1	45.00%
requirement 1	70.00%
task 1	70.00%
task 2	75.00%
Capability 2	55.00%
requirement 2	60.00%
task 2	75.00%
task 3	45.00%
requirement 3	60.00%
task 2	75.00%
task 3	45.00%
task 4	40.00%

OC-2 finished WI-201 : Mobile Team : : task 1

Figure 1. Prototype Player

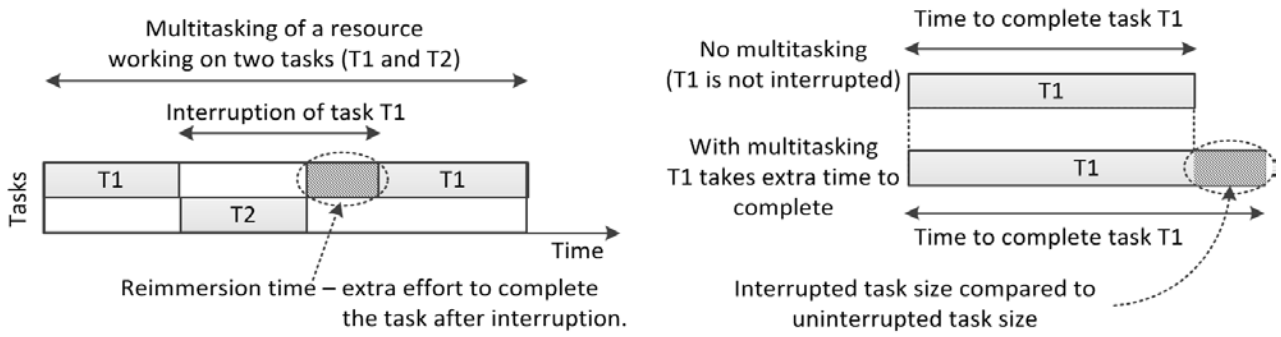

Multitasking of a resource working on two tasks (T1 and T2)
Interruption of task T1
Tasks
T1
T1
T2
Time
Reimmersion time – extra effort to complete the task after interruption.
No multitasking (T1 is not interrupted)
Time to complete task T1
T1
With multitasking T1 takes extra time to complete
T1
Time to complete task T1
Interrupted task size compared to uninterrupted task size

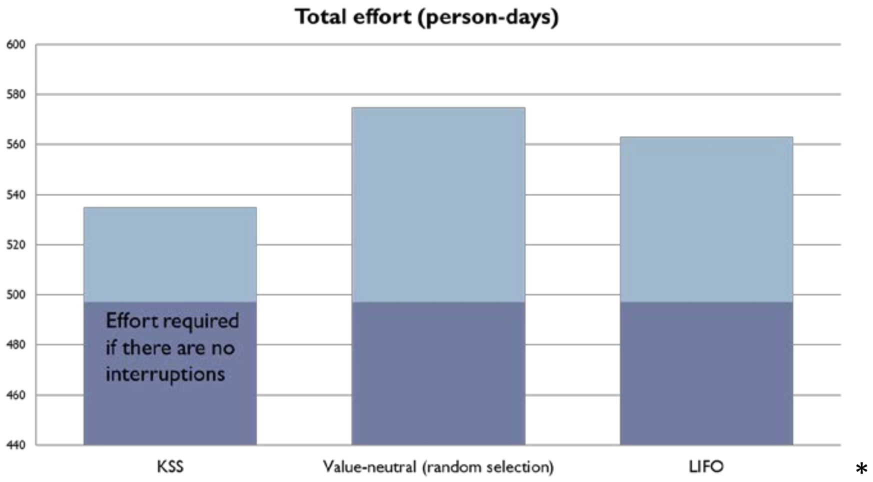
Figure 2. Reimmersion time.

DATASEM enabled the use of reimmersion time to model the impact of work interruptions in simulation.

The following simulation scenario was used:

- 10 teams (20 members each) + system engineering team.
- 20 new capabilities at start.
- Each capability unfolds into 30 requirements on average
- Each requirement unfolds into 9 tasks on average.
- Each task takes 3-15 days.
- Simulation time-frame: 1 day.
- Reimmersion time: 30 minutes

The simulation results show for value-neutral work prioritization impact of work interruptions can be as highs as 15% of the overall effort (Figure 3). These results are consistent with work log observations of industry projects where among 6 projects that value was between 14-15.5% [10].



*KSS here means value-based work prioritization.

Figure 3. The impact of work interruptions with prioritization strategies.

NEXT STEPS

There are many opportunities for further research on adaptive processes in systems engineering, operational system evolution, and system/product development. The first task, because it is the primary enabler for most of the research opportunities, is to conduct the definition review survey with industry and government experts, using the information to inform, revise, validate and complete the DATASEM implementation. Once that is accomplished, here are the most promising opportunities for further DATASEM research:

- Understand the impact of work item characteristics for both adaptive and predictive scheduling; determine if planning horizon and batch size are the key considerations.
- Implement human characteristics and culture in the organizational components; model service definition and negotiation in the acceptance model to understand how both culture and governance play a role.
- Adapt DATASEM to work with the Experience Accelerator to model adaptive systems engineering and development environments; this will allow learners to experience the differences, strengths and weaknesses of various combinations of adaptive and predictive organizational components.
- Build a library of experiments, model templates and data based on industry experience as well as proposed results to accelerate the use of DATASEM in research and operations; determine the most effective way to organize a community of DATASEM users and provide a custodian to manage its continuing evolution.
- Continue evolution of the Analysis Tool to take advantage of the library and provide additional functionality to compare new experiments with those in the library.
- Use the data and experiments to develop a systems dynamics model that represents the DATASEM outcomes in terms of mathematical functions rather than discrete work items.

REFERENCES

[1] Dahmann, J.S.; Baldwin, K.J., "Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering," *Systems Conference, 2008 2nd Annual IEEE* , vol., no., pp.1,7, 7-10 April 2008

[2] Boehm, B; Koolmanojwong, S.; Lane, J; Turner, R.; “Principles for Successful Systems Engineering,” Conference on Systems Engineering Research, Mar 19-22, 2012, Procedia Computer Science, Elsevier, Vol. 8, 2012

[3] Larman C. and Vodde, B. (2009). *Scaling Lean & Agile Development.* Boston, MA: Addison Wesley.

[4] Anderson, David. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business*. Sequim, WA: Blue Hole Press

[5] Reinertsen, Donald G. (2010). *The Principles of Product Development Flow*. Redondo Beach, CA: Celeritas Publishing.

[6] Turner, R.; “A Lean Approach to Scheduling Systems Engineering Resources,” CrossTalk, May/June, 2013.

[7] Turner, R.; Madachy, R.; Lane, J.A.; Ingold, D.; Levine, L.; “Agile-Lean Software Engineering (ALSE): Evaluating Kanban in Systems Engineering,” A013 - Final Technical Report SERC-2013-TR-022-2, March 6, 2013.

[8] Lane, J. A. and Turner, R. “Improving Development Visibility and Flow in Large Operational Organizations,” 4th International Conference on Lean Enterprise Software and Systems, Galway, Ireland, December 1-4, 2013, Proceedings, Lecture Notes in Business Information Processing, Vol. 167, pp 65-80, Springer-Verlag, Heidelberg, 2013.

[9] DeMarco, T. and Lister, T., 2013. Peopleware: productive projects and teams. Addison-Wesley.

[10] Tregubov, Alexey, Lane, J.A., Boehm B., "Evaluation of cross-project multitasking in software projects" Conference on Systems Engineering Research (CSER’17), 23-25 March 2017.

APPENDIX A: MODELING SE ACTIVITIES IN SIMIO AND COMPARING SIMULATION RESULTS WITH DATASEM DSL/REPAST MODEL

INTRODUCTION

SIMIO is an Object-oriented programming platform under a Discrete-Event Simulation (DES) framework. Users build their models by utilizing a variety of intelligent objects, elements, processes and steps provided in SIMIO's libraries.

Repast is based on Agent-Based Simulation (ABS) and is the basis for the DATASEM Simulator program, developed in Java. Modeling is done via the DATASEM Modeler DSL program.

As modeling and simulation frameworks DES and ABS have inherently different underlying mechanisms. However, the simulation results should be the same given the same experiment inputs, specifically the Organizational Model, Work Item Network Model, Governance Strategies and other experimental parameters. Modeling and simulating in different platforms helps us to cross-validate our conceptual models and helps the team reach agreement on the key DATASEM governance concepts.

The organizational model (Figure 4) is implemented in SIMIO model. The same model is coded in DSL.

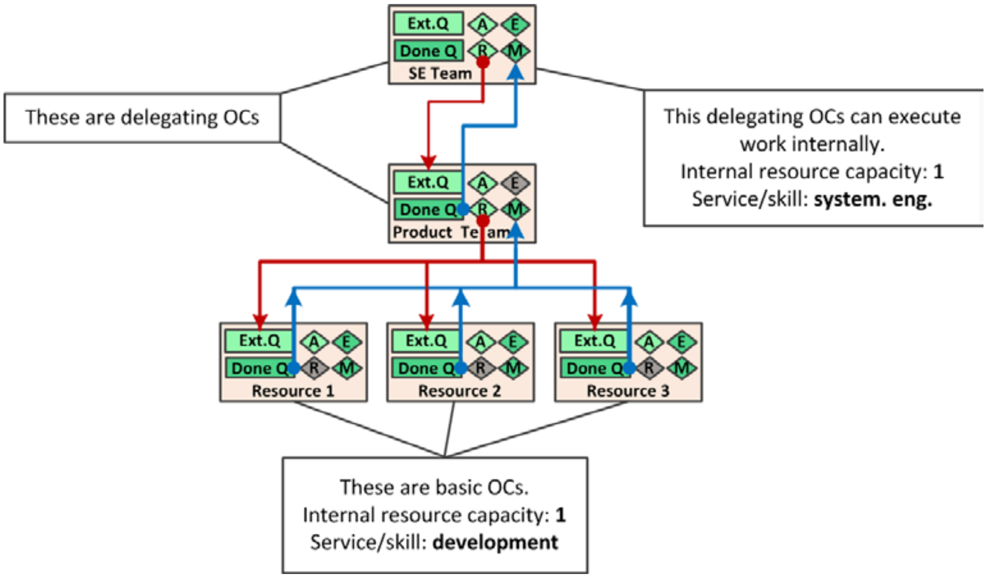

Ext.Q
Done Q
A
E
R
M
SE Team
These are delegating OCs
This delegating OCs can execute work internally.
Internal resource capacity: **1**
Service/skill: **system. eng.**
Product Team
Resource 1
Resource 2
Resource 3
These are basic OCs.
Internal resource capacity: **1**
Service/skill: **development**

Figure 4. Organizational model

Figure 5 shows a Work Item Network Model (described below) imported into the SIMIO program. The same model was coded into DSL.

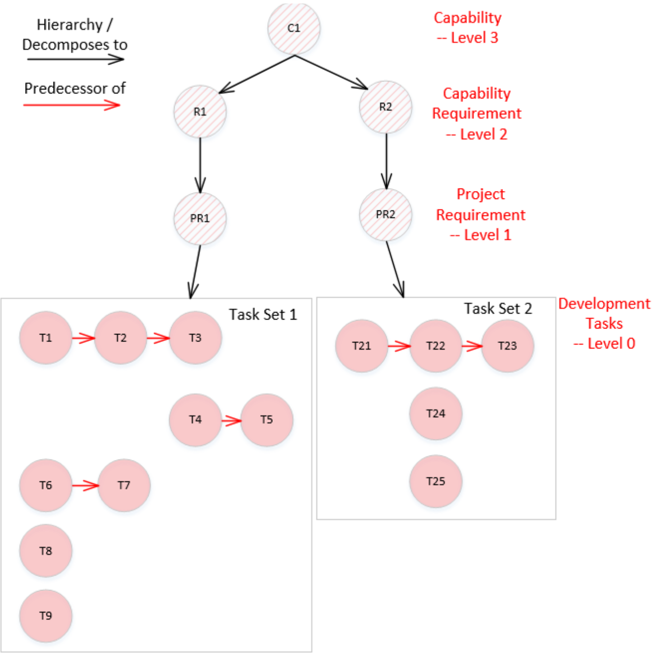

Hierarchy / Decomposes to
Predecessor of
C1
Capability -- Level 3
R1
R2
Capability Requirement -- Level 2
PR1
PR2
Project Requirement -- Level 1
Task Set 1
Task Set 2
Development Tasks -- Level 0
T1
T2
T3
T4
T5
T6
T7
T8
T9
T21
T22
T23
T24
T25

Figure 5. Work Item Network Model

Additionally, each Work Item is assigned a predefined "Value" – 1, 2 … 9 for T1, T2 … T9, 1.1, 2.1 … 5.1 for T21, T22 … T25, and the same value-based prioritization is applied so that the results of both models are expected to be deterministic and, by definition, the same.
The SIMIO model Facility view of a Product Development Team is provided in Figure 6. Numbers on the tag above each task shows its Perceived Value when simulation is running):

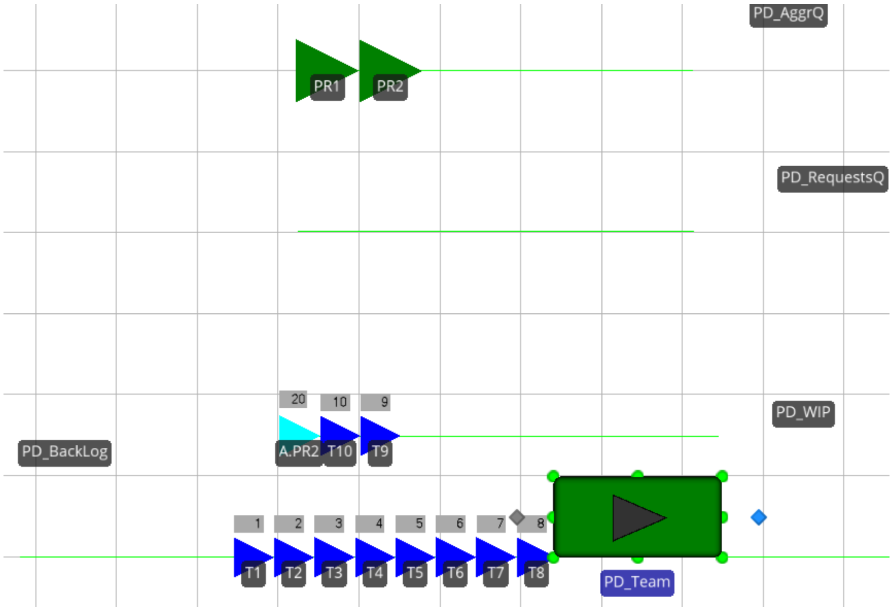
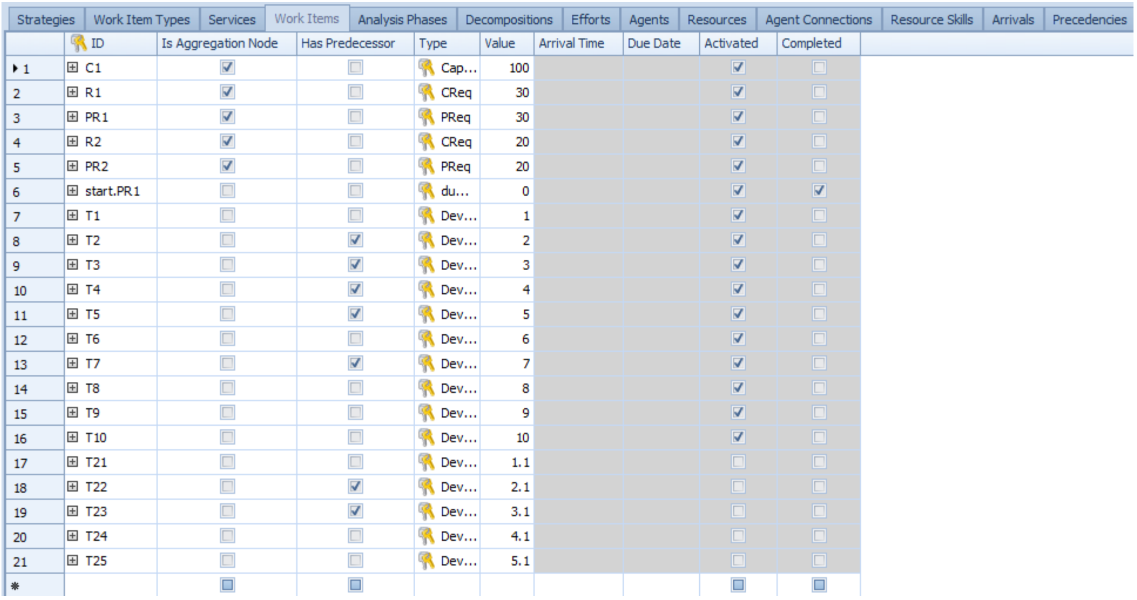


Figure 6. SIMIO Product Development Team representation

The data structure shown in Figure 7 is consistent with that of the DSL program, so direct importing of DSL to SIMIO is possible.



	ID	Is Aggregation Node	Has Predecessor	Type	Value	Arrival Time	Due Date	Activated	Completed
▸ 1	⊞ C1	☑	□	Cap...	100			☑	□
2	⊞ R1	☑	□	CReq	30			☑	□
3	⊞ PR1	☑	□	PReq	30			☑	□
4	⊞ R2	☑	□	CReq	20			☑	□
5	⊞ PR2	☑	□	PReq	20			☑	□
6	⊞ start.PR1	□	□	du...	0			☑	☑
7	⊞ T1	□	□	Dev...	1			☑	□
8	⊞ T2	□	☑	Dev...	2			☑	□
9	⊞ T3	□	☑	Dev...	3			☑	□
10	⊞ T4	□	☑	Dev...	4			☑	□
11	⊞ T5	□	☑	Dev...	5			☑	□
12	⊞ T6	□	□	Dev...	6			☑	□
13	⊞ T7	□	☑	Dev...	7			☑	□
14	⊞ T8	□	□	Dev...	8			☑	□
15	⊞ T9	□	□	Dev...	9			☑	□
16	⊞ T10	□	□	Dev...	10			☑	□
17	⊞ T21	□	□	Dev...	1.1			□	□
18	⊞ T22	□	☑	Dev...	2.1			□	□
19	⊞ T23	□	☑	Dev...	3.1			□	□
20	⊞ T24	□	□	Dev...	4.1			□	□
21	⊞ T25	□	□	Dev...	5.1			□	□
*		□	□					□	□

Figure 7. SIMIO Data Table for WIN information

SIMULATION OUTPUT COMPARISON

SIMIO

Figure 8 shows the SIMIO generated resource allocation status along with time. The length of each block shows the actual **Cycle Time** of each task (from start to completion).

In Figure 9, the length of each block shows the actual **Lead Time** of each task (from accepted till completed) at each team.

The **Gantt chart** in

Figure 10 shows the team service & resource allocation status on each task.

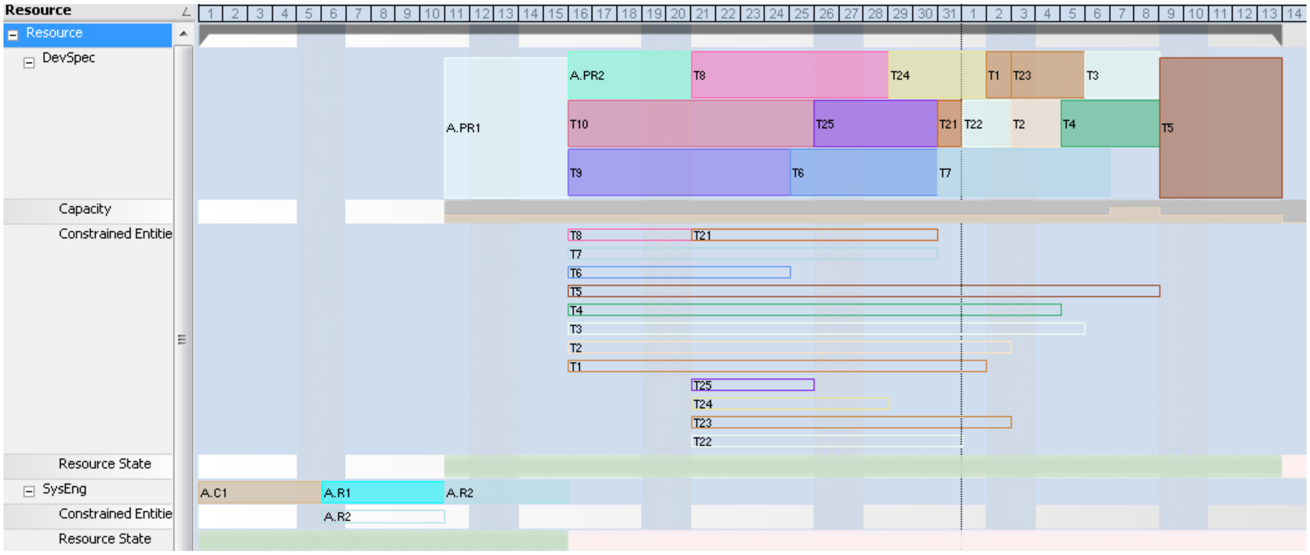


Figure 8. Resource Allocation status on SIMIO Resource Planning and Scheduling interface

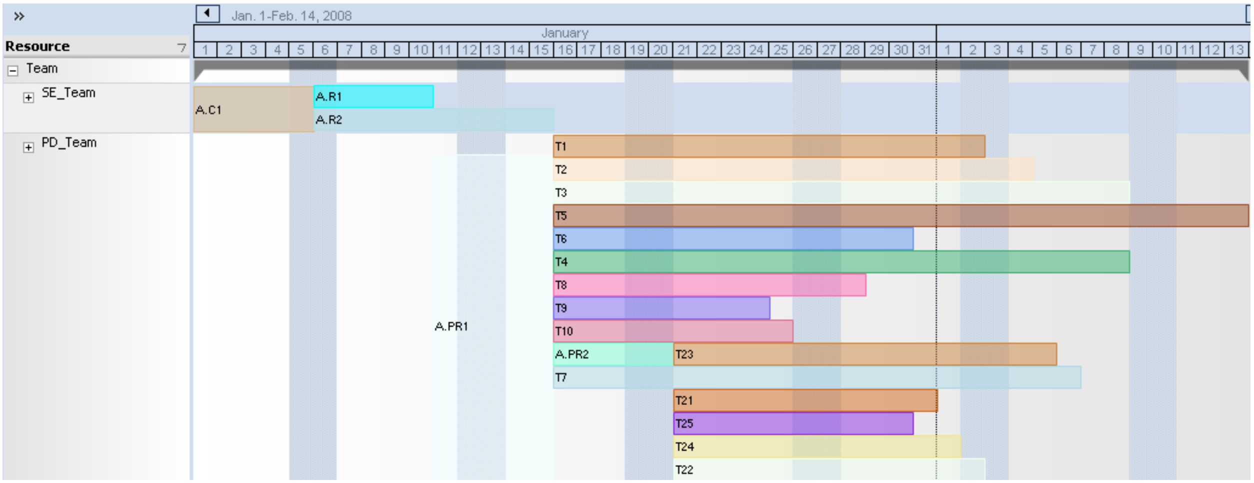


Figure 9. Lead Time in SIMIO

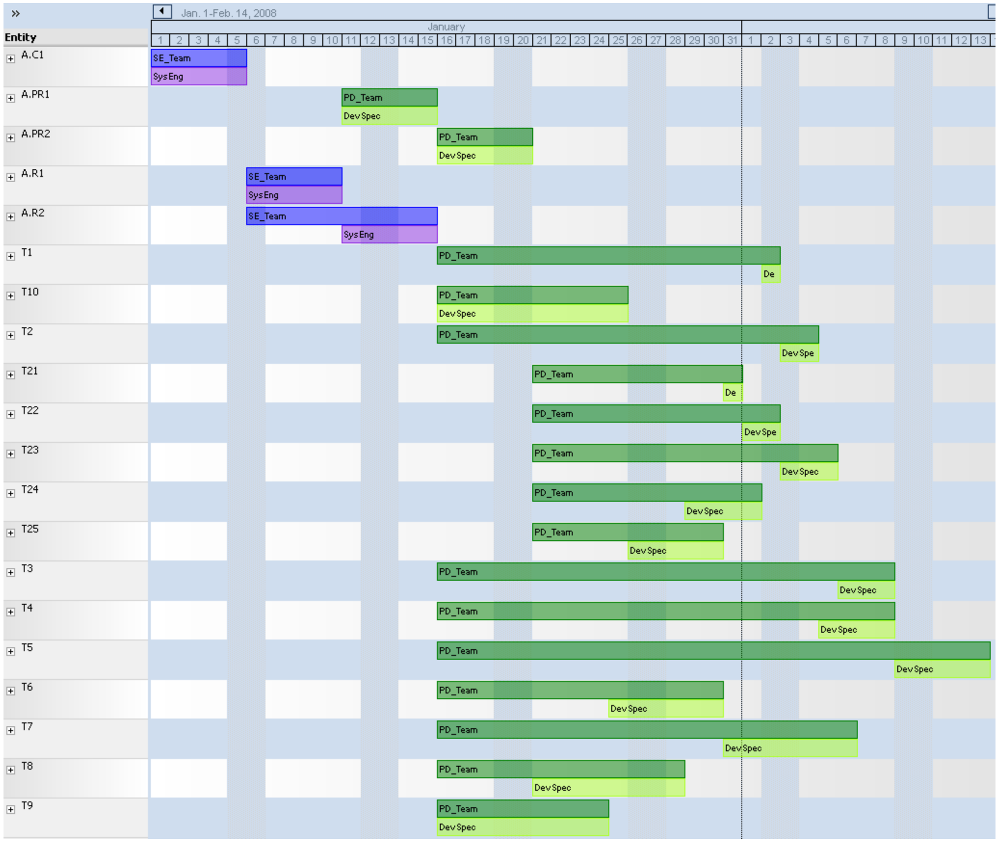


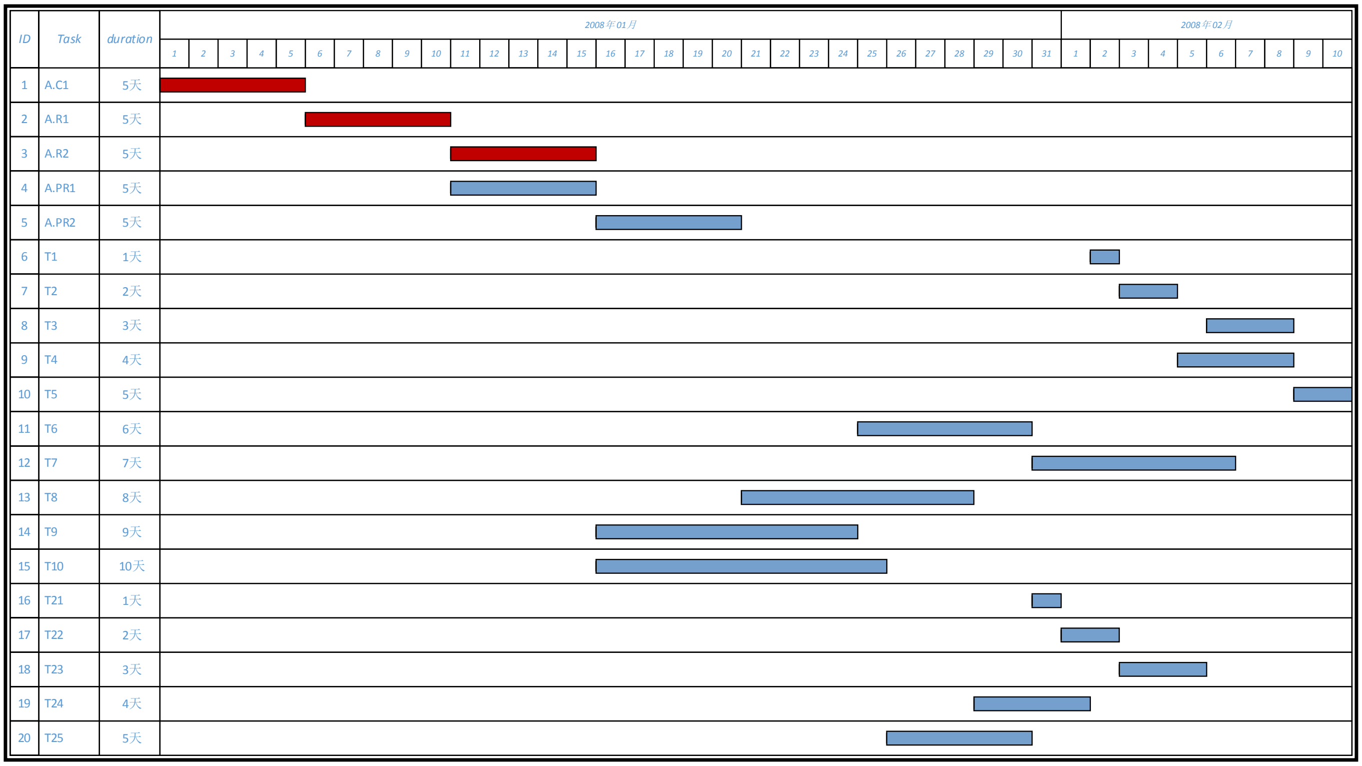
Figure 10. Entity (Work Item) Flow status

DSL / REPAST

Figure 11 shows the Work Item flow manually created from the DATASEM Simulator (Repast) log records.

RESULTS

Both cases applied value-based work prioritization, with the perceived value predefined. The generated schedules are same in both tools, and completion duration for capability C1 (and all the tasks decomposed along all 4 levels) is 44 days. Precedence constraints are not violated.



ID	Task	duration
1	A.C1	5天
2	A.R1	5天
3	A.R2	5天
4	A.PR1	5天
5	A.PR2	5天
6	T1	1天
7	T2	2天
8	T3	3天
9	T4	4天
10	T5	5天
11	T6	6天
12	T7	7天
13	T8	8天
14	T9	9天
15	T10	10天
16	T21	1天
17	T22	2天
18	T23	3天
19	T24	4天
20	T25	5天

Figure 11. DATASEM Execution Gantt Chart

DATASEM Simulator (Repast) log:

```
REPLICATION #1
START @TIME:1 Capability[Hierarchy:3]C1(id:1) is Started
COMPLETION @TIME:5 Analysis[Hierarchy:0]Ana.0-C1(id:21) is Completed
START @TIME:6 CReq[Hierarchy:2]R1(id:2) is Started
COMPLETION @TIME:10 Analysis[Hierarchy:0]Ana.0-R1(id:22) is Completed
Contract Built
Manager:SET01, Contractor:PDT01, on:
 Analysis[Hierarchy:0]Ana.0-PR1(id:24)
START @TIME:11 CReq[Hierarchy:2]R2(id:4) is Started
START @TIME:11 PReq[Hierarchy:1]PR1(id:3) is Started
COMPLETION @TIME:15 Analysis[Hierarchy:0]Ana.0-R2(id:23) is Completed
COMPLETION @TIME:15 Analysis[Hierarchy:0]Ana.0-PR1(id:24) is Completed
Contract Built
Manager:SET01, Contractor:PDT01, on:
 Analysis[Hierarchy:0]Ana.0-PR2(id:25)
START @TIME:16 DevTask[Hierarchy:0]T10(id:15) is Started
START @TIME:16 DevTask[Hierarchy:0]T9(id:14) is Started
START @TIME:16 PReq[Hierarchy:1]PR2(id:5) is Started
COMPLETION @TIME:20 Analysis[Hierarchy:0]Ana.0-PR2(id:25) is Completed
START @TIME:21 DevTask[Hierarchy:0]T8(id:13) is Started
COMPLETION @TIME:24 DevTask[Hierarchy:0]T9(id:14) is Completed
START @TIME:25 DevTask[Hierarchy:0]T6(id:11) is Started
COMPLETION @TIME:25 DevTask[Hierarchy:0]T10(id:15) is Completed
START @TIME:26 DevTask[Hierarchy:0]T25(id:20) is Started
COMPLETION @TIME:28 DevTask[Hierarchy:0]T8(id:13) is Completed
START @TIME:29 DevTask[Hierarchy:0]T24(id:19) is Started
COMPLETION @TIME:30 DevTask[Hierarchy:0]T6(id:11) is Completed
COMPLETION @TIME:30 DevTask[Hierarchy:0]T25(id:20) is Completed
START @TIME:31 DevTask[Hierarchy:0]T7(id:12) is Started
START @TIME:31 DevTask[Hierarchy:0]T21(id:16) is Started
COMPLETION @TIME:31 DevTask[Hierarchy:0]T21(id:16) is Completed
START @TIME:32 DevTask[Hierarchy:0]T22(id:17) is Started
COMPLETION @TIME:32 DevTask[Hierarchy:0]T24(id:19) is Completed
START @TIME:33 DevTask[Hierarchy:0]T1(id:6) is Started
COMPLETION @TIME:33 DevTask[Hierarchy:0]T22(id:17) is Completed
COMPLETION @TIME:33 DevTask[Hierarchy:0]T1(id:6) is Completed
START @TIME:34 DevTask[Hierarchy:0]T2(id:7) is Started
START @TIME:34 DevTask[Hierarchy:0]T23(id:18) is Started
COMPLETION @TIME:35 DevTask[Hierarchy:0]T2(id:7) is Completed
START @TIME:36 DevTask[Hierarchy:0]T4(id:9) is Started
COMPLETION @TIME:36 DevTask[Hierarchy:0]T23(id:18) is Completed
COMPLETION @TIME:36 PReq[Hierarchy:1]PR2(id:5) is Completed
COMPLETION @TIME:36 CReq[Hierarchy:2]R2(id:4) is Completed
START @TIME:37 DevTask[Hierarchy:0]T3(id:8) is Started
COMPLETION @TIME:37 DevTask[Hierarchy:0]T7(id:12) is Completed
COMPLETION @TIME:39 DevTask[Hierarchy:0]T4(id:9) is Completed
COMPLETION @TIME:39 DevTask[Hierarchy:0]T3(id:8) is Completed
START @TIME:40 DevTask[Hierarchy:0]T5(id:10) is Started
COMPLETION @TIME:44 DevTask[Hierarchy:0]T5(id:10) is Completed
COMPLETION @TIME:44 PReq[Hierarchy:1]PR1(id:3) is Completed
COMPLETION @TIME:44 CReq[Hierarchy:2]R1(id:2) is Completed
COMPLETION @TIME:44 Capability[Hierarchy:3]C1(id:1) is Completed
SIMULATION ENDED: All WIs Completed
EndTime: 44
Value Delivered: 270.5
NPV: 270.5
```

APPENDIX B: INITIAL OUTPUT INDICATORS (JSON FORMAT DESCRIPTION)

The following tables describe the current output indicators model to represent data that could be displayed via simulation player. The JSON format has three sections which describe these indicators/diagrams:

1. oc_dictionary - a list of indicators associated with each OC
2. work_item_dictionary - a list of indicators associated with each WI
3. frame_dictionary - a list of indicators not associated with individual OC or WI

These dictionaries are used to gather and format indicator information for display. They are then read into the player software and provide time-specific status in charts, gauges, graphics, and a textual log of simulation events (e.g. work accepted, work completed).

Data		
Property Name	Format	Description
oc_dictionary	[indicator_1,indicator_2, ...]	Organization indicators dictionary
work_item_dictionary	[indicator_1, indicator_2, ...]	Work Item indicators dictionary
event_dictionary	[indicator_1, indicator_2, ...]	Event indicators dictionary
frame_dictionary	[indicator_1, indicator_2, ...]	Frame indicators dictionary
frames	[frame_1, frame_2, frame_3, ...]	Frame list.

Indicator (for oc_dictionary, work_item_dictionary, event_dictionary, frame_dictionary)		
Property Name	Format	Description
name	“done”	Name of indicator
x	“Time”	X-axis name
y	“#”	Y-axis name
title	“# of Completed Work Items”	Title of indicator

Frames		
Property Name	Format	Description
organization_components	[oc_1, oc_2, ...]	Organization list
work_items	[wi_1, wi_2, ...]	Work Item list.
events	[event_1, event_2, …]	Event list.
aggregating_indicators	[{“wip”: 10}, {“done”: 20}, ...]	List of aggregating indicators of frame

Organization Component		
name	“Tom”	Name
description	“Example Team”	Description
id	“123”	ID of Organization Component.
skills	[{name:“C”, proficiency:”1.5”}, ...]	Skills and proficiency of Organization Component.
external_queue	[“201”, “202”, “203”, ...]	External Work Item queue of Organization Component.
accepted_queue	[“201”, “202”, “203”, ...]	Accepted Work Item queue
working_queue	[“201”, “202”, “203”, ...]	Queue of Work Items which Organization Component is working on.
indicators	[10, 0.1, ...]	Values of Indicators for Organization Component, as described by oc_dictionary.

Work Item		
id	“201”	ID of Work Item
type	“Capability”	Type of Work Item.
name	“Task_1”	Name of Work Item
description	“This is Task_1”	Description of Work Item
skills	[string, string, ….]	List of skills that are needed to complete this Work Item.
assigned_to	“1”	Assigned Organization Component
indicators	[101, 10.0]	List of indicators descripted by wi_dictionary.
children	[“1”, “2”, “3”]	List of Children Work Items

Event		
src_oc_id	“1”	Source OC id
dst_oc_id	“2”	Destination OC id
work_item_id	“201”	Work Item Id
type	“WI_Delegation”	Type of event
description	“...”	description
indicators	[“indicator_1”:11, “indicator_2”:12,...]	List of indicators.